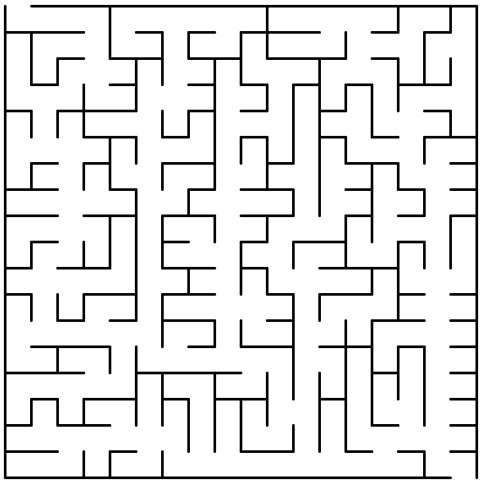
MAZE: A Secure Cloud Storage Service Using Moving Target Defense and Secure Shell Protocol (SSH) Tunneling

Vasco Xu, Sherif Khattab
University of Pittsburgh, USA

University of Pittsburgh

MAZE



Agenda

- Introduction
- Threat Model and Assumptions
- MAZE Overview and Design
- MAZE Experiments
- Limitations and Future Work

Introduction

- Cloud storage systems are rising in popularity
  - Increase in online collaboration
  - Promise to be accessible anytime, anywhere
  - Deloitte reports that 58% percent of a total of 500 IT leaders moved to the cloud because of security and data protection
- Cloud storage security
  - Gartner Inc. estimates that 95% of cloud breaches are due to human errors

PRO CYBER NEWS

Human Error Often the Culprit in Cloud Data Breaches

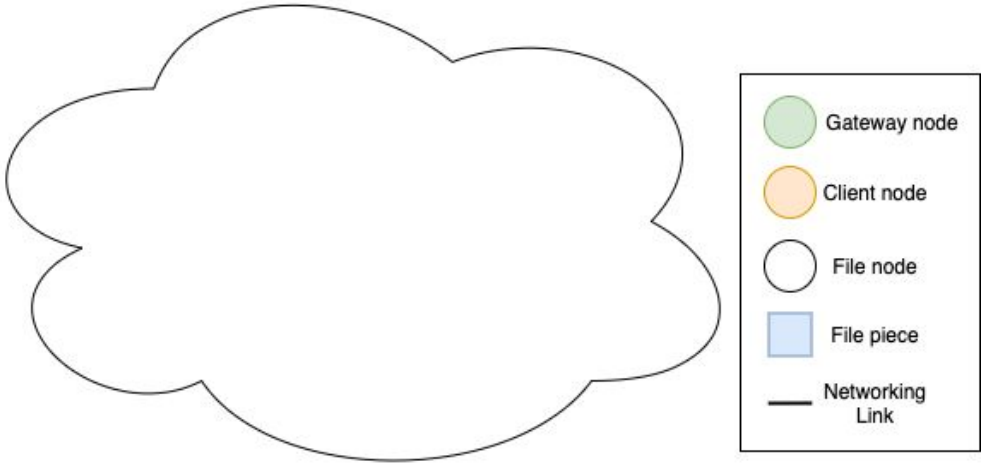
Mistakes made by customers can often lead to the finger being pointed at cloud providers

Gartner Inc. estimates that up to 95% of cloud breaches occur due to human errors such as configuration mistakes, and the research firm expects this trend to continue.

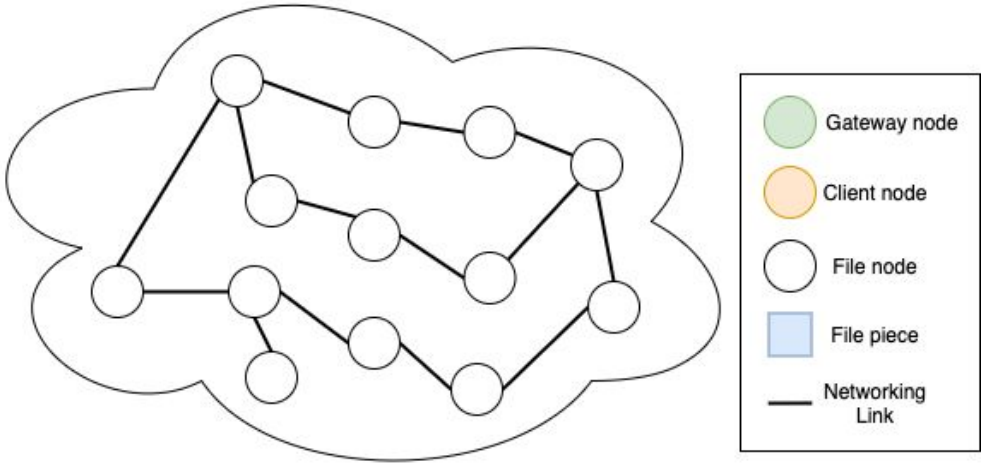
Moving Target Defense (MTD)

- **However**, cloud storages are static attack targets
  - Attackers freely explore the system
  - Static defense mechanisms
- MTD increases difficulty and cost of executing attacks
  - Randomize attack surface
  - Attackers face a greater deal of uncertainty
- Scalable cybersecurity solution
  - Reduce the need for threat detection
  - Lessen the burden on threat detection software and cybersecurity teams

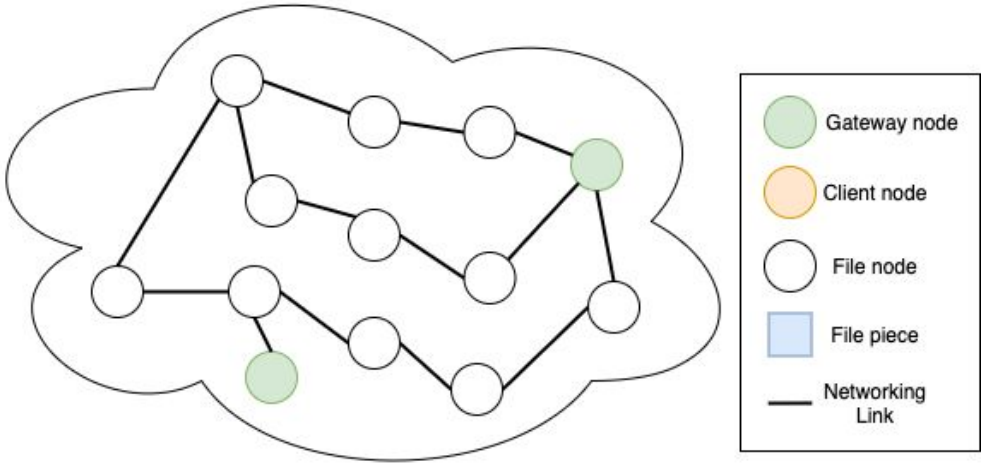
System Model



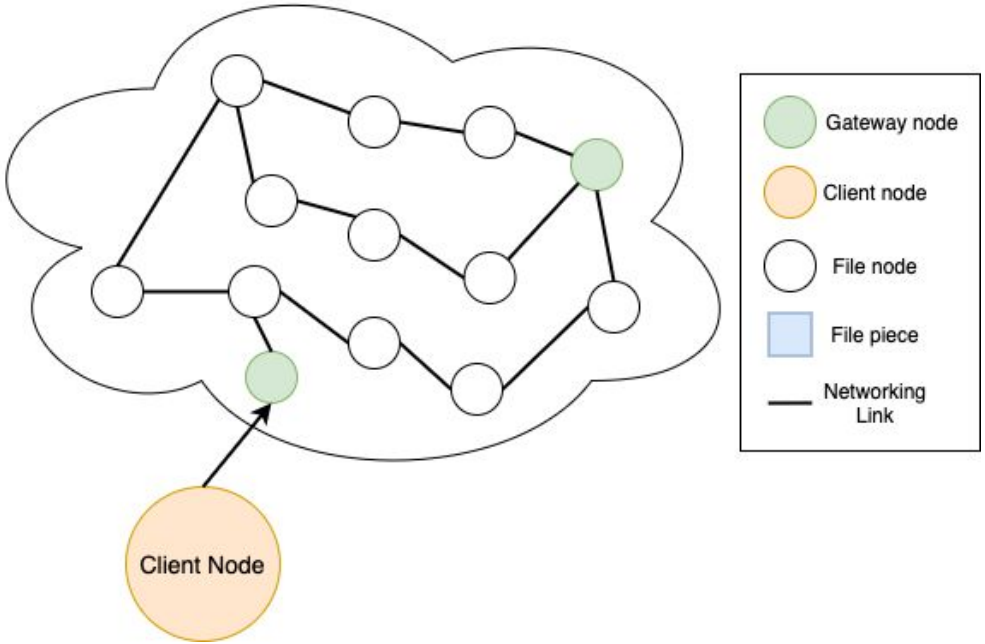
System Model



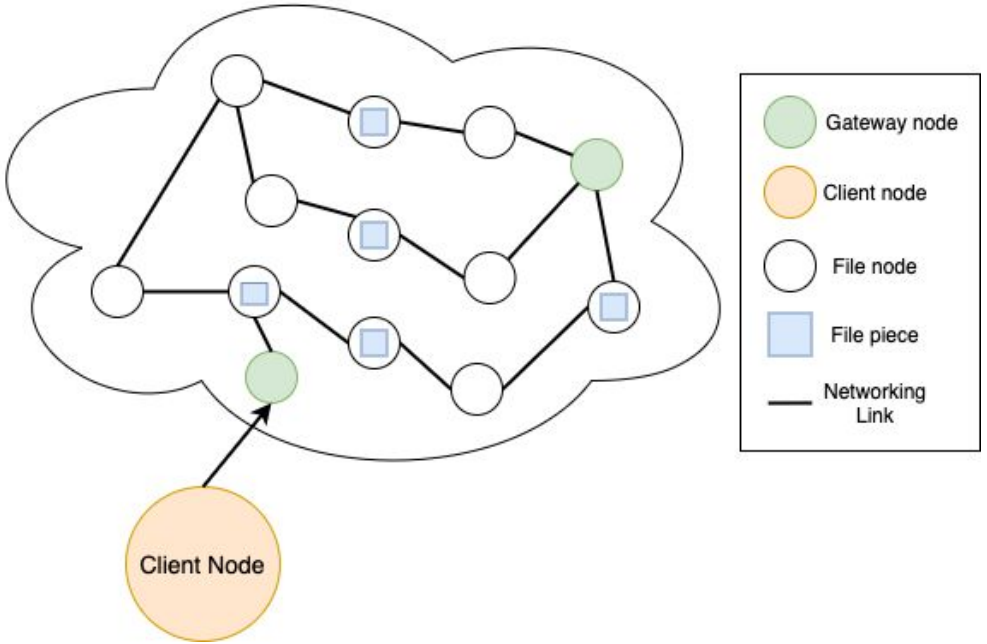
System Model



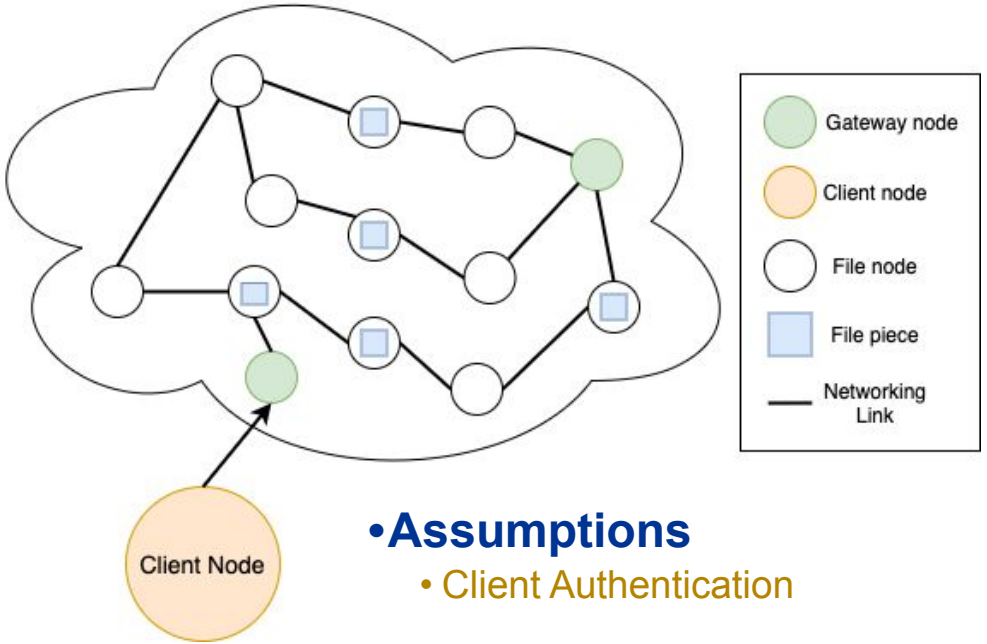
System Model



System Model

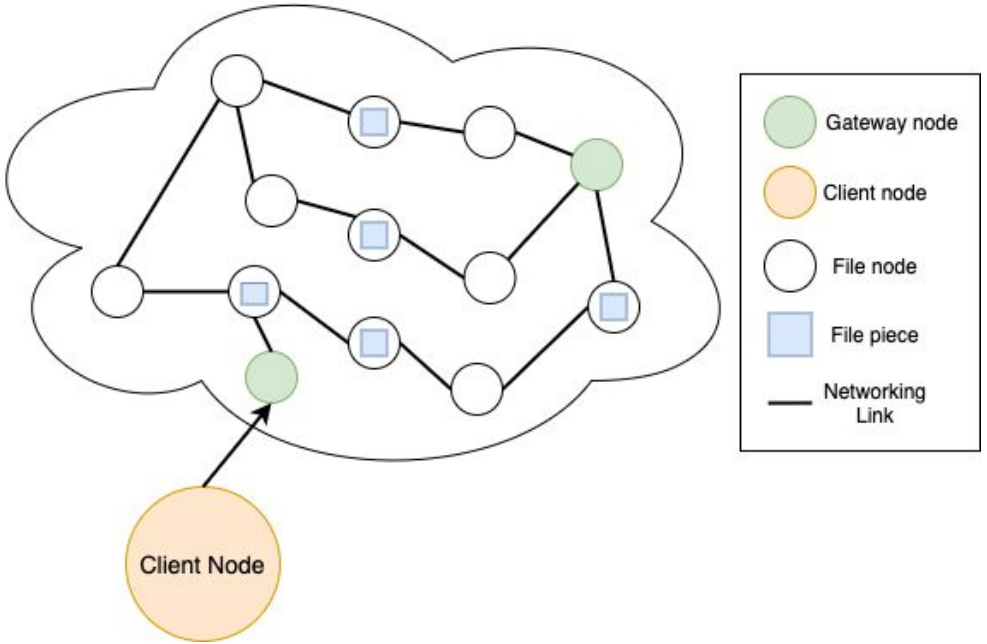


System Model

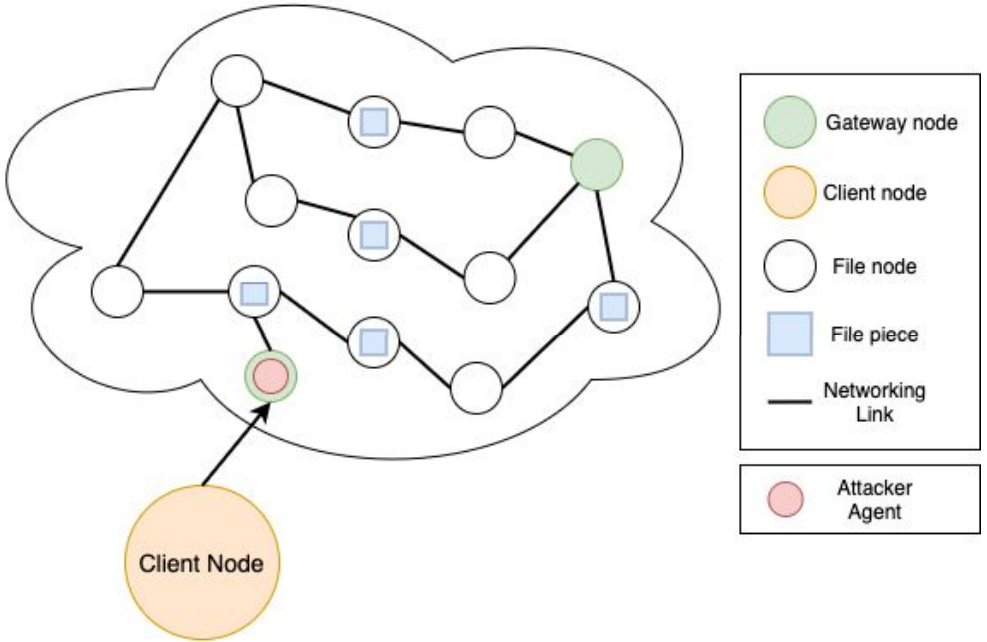


- **Assumptions**
  - Client Authentication

System Model



System Model



Threat Model

- **Attacker**
  - **Goal:** Recover a specific file
  - **Success:** Retrieves all file pieces

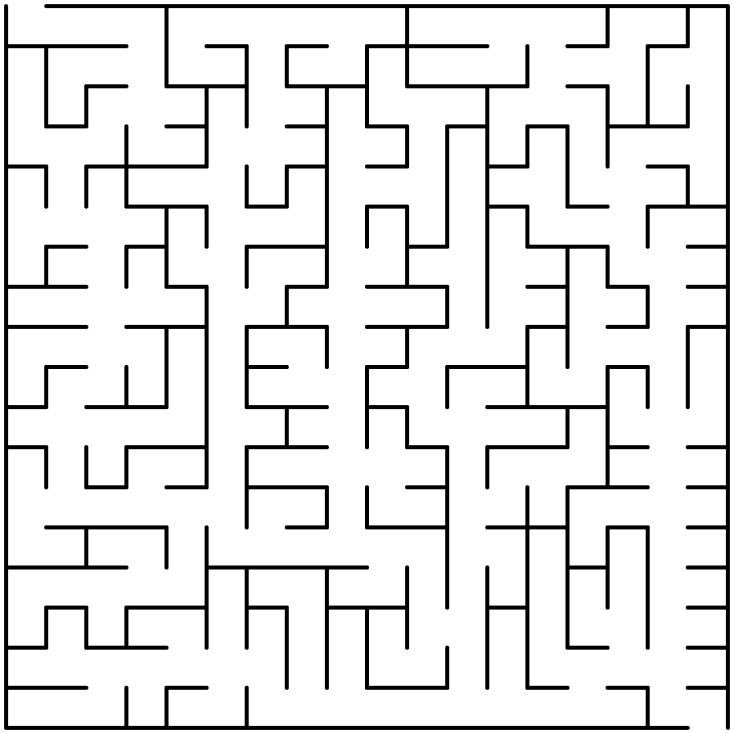
Assumptions on the Attacker

- Attackers can retrieve any file by taking control over a growing subset of nodes
  - Take control of nodes incrementally
- Attackers do not know how many pieces a file was split into
  - Full traversal to retrieve all possible file pieces
- Attackers are not authorized users
  - Cannot break authentication or steal client credentials or access keys

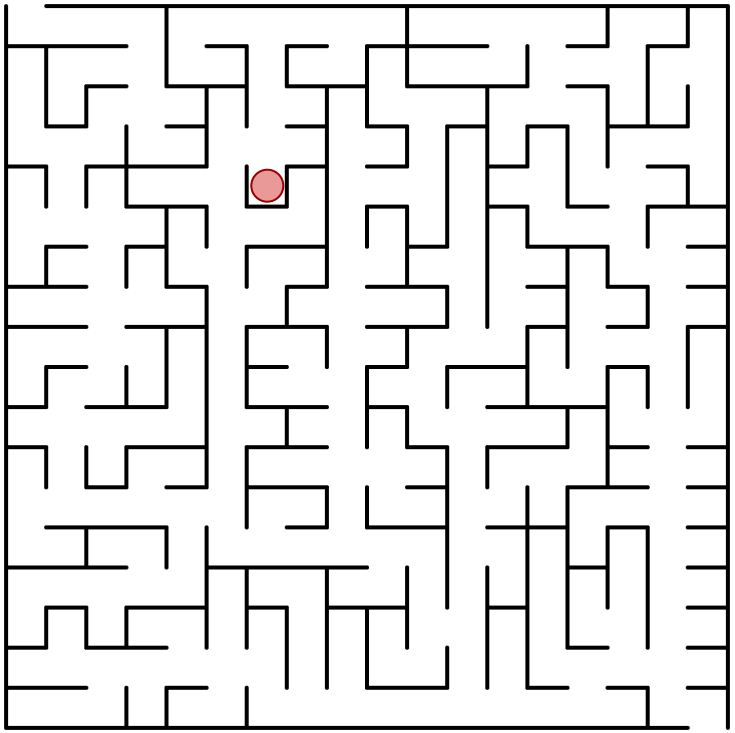
Problem Statement

- Build a secure cloud storage under the following constraints:
  - **Defense perspective:**
    - Given a number of malicious agents and a specific time limit, probability of retrieving all file pieces is very low.
  - **Performance perspective:**
    - Given the described system model, build a secure cloud storage that provides adequate response time in addition to security.

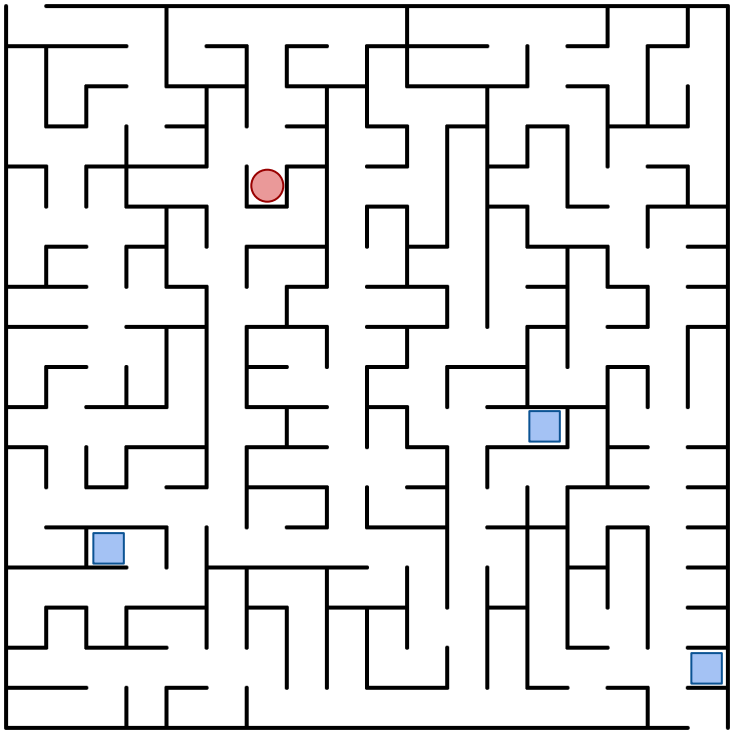
MAZE Overview



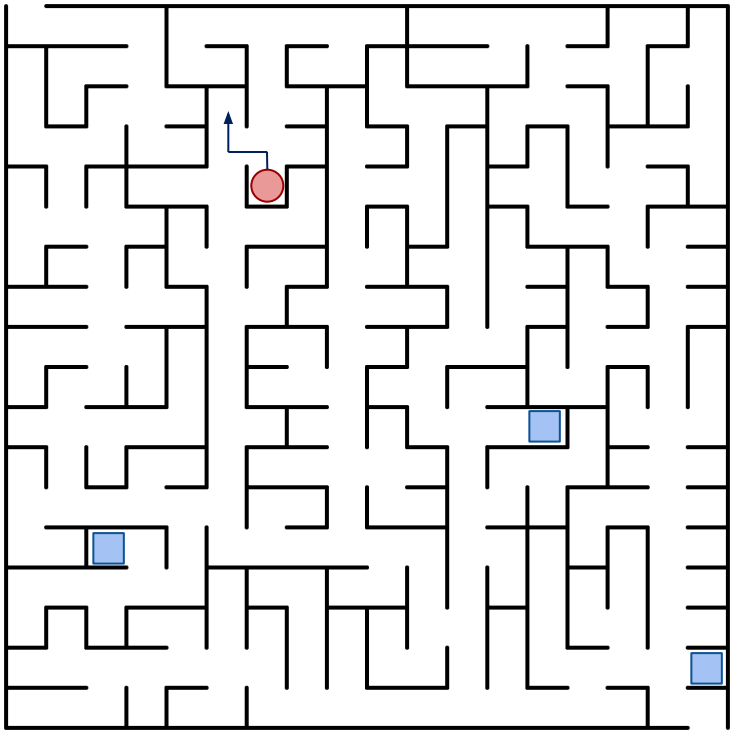
MAZE Overview



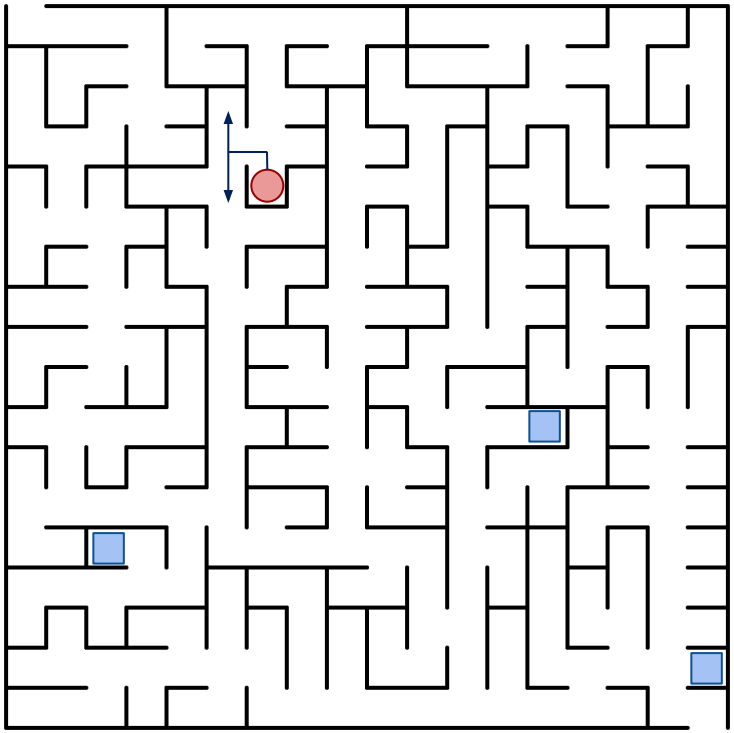
MAZE Overview



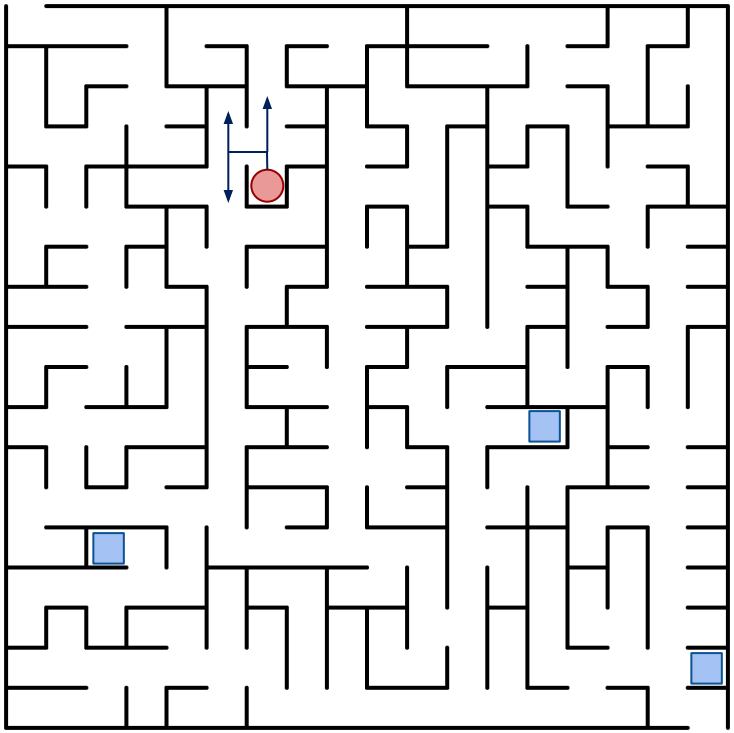
MAZE Overview



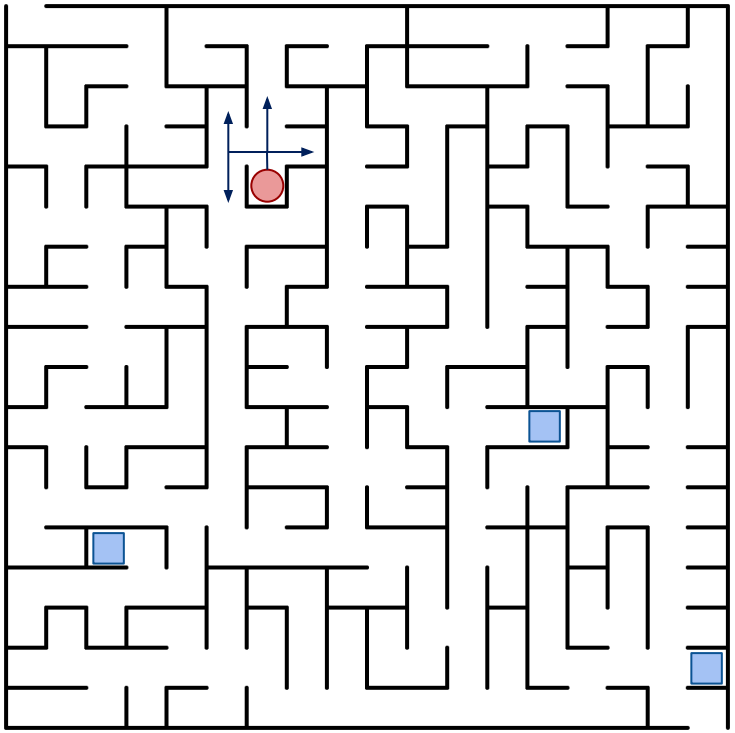
MAZE Overview



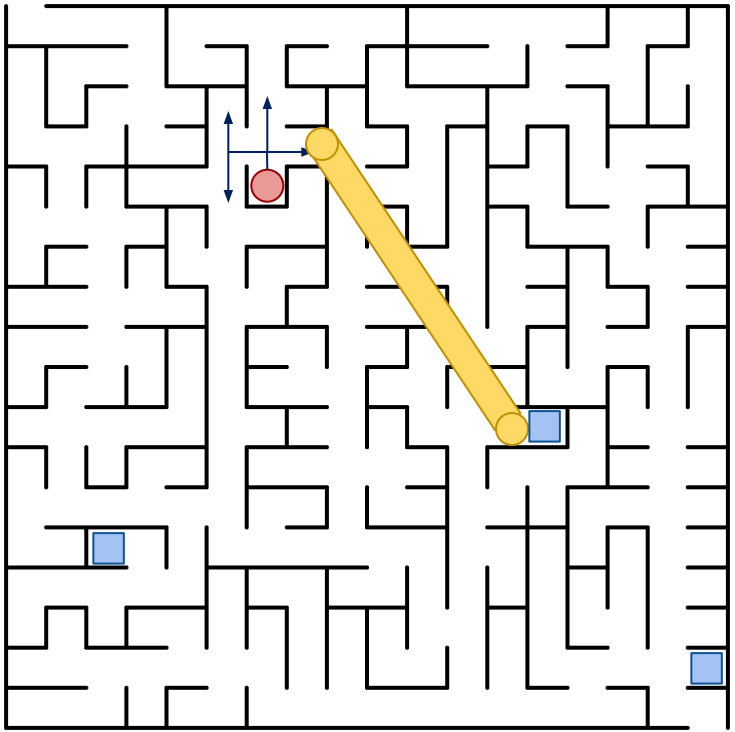
MAZE Overview



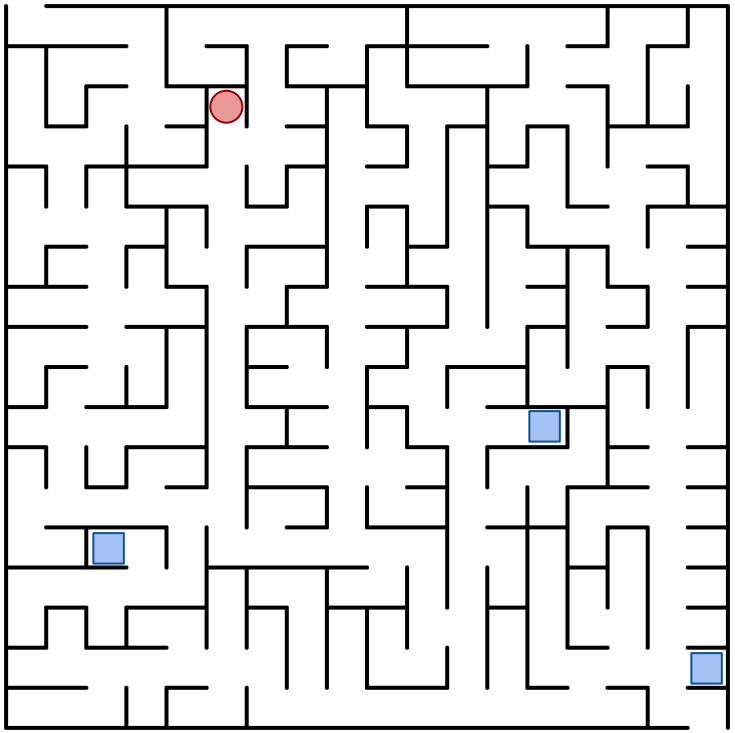
MAZE Overview



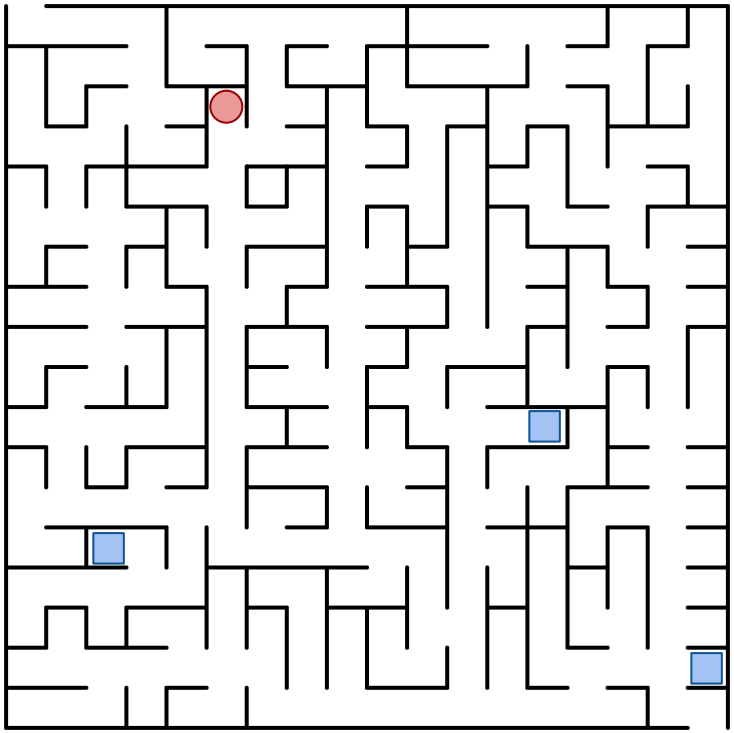
MAZE Overview



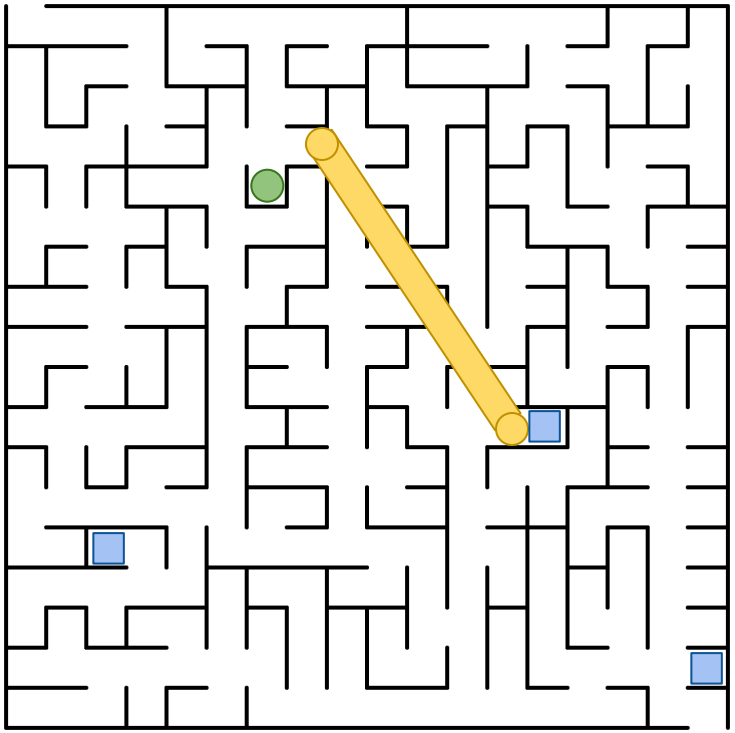
MAZE Overview



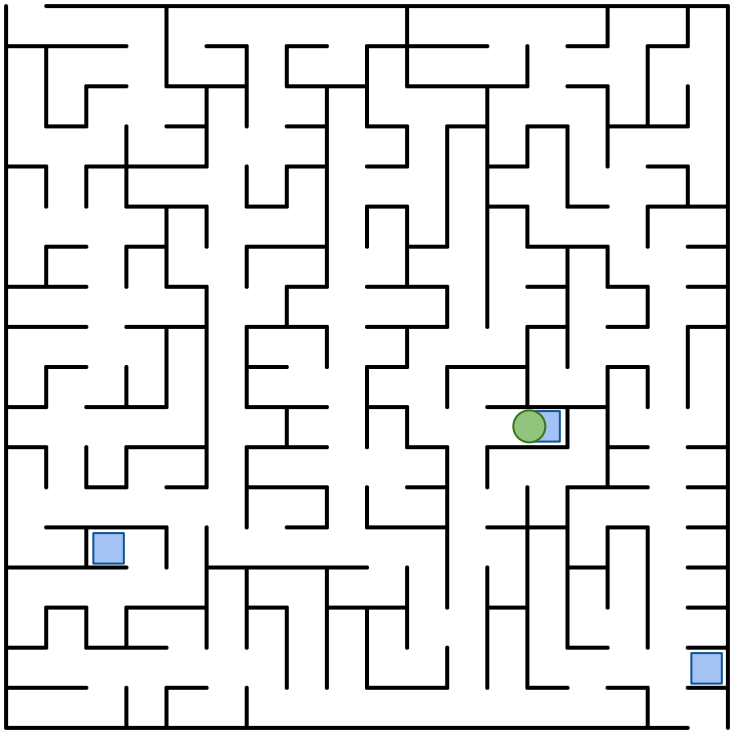
MAZE Overview



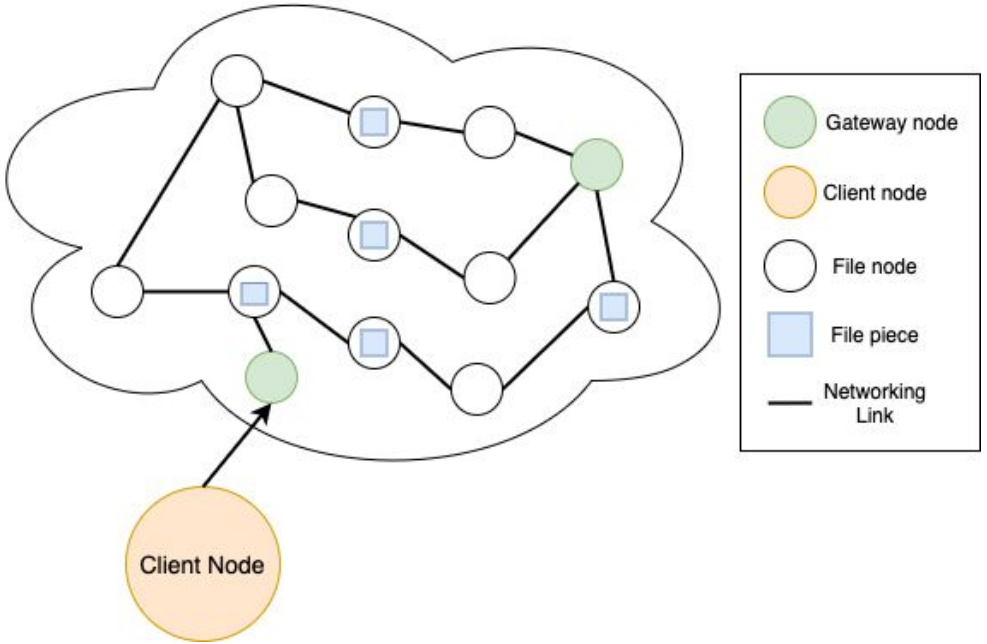
MAZE Overview



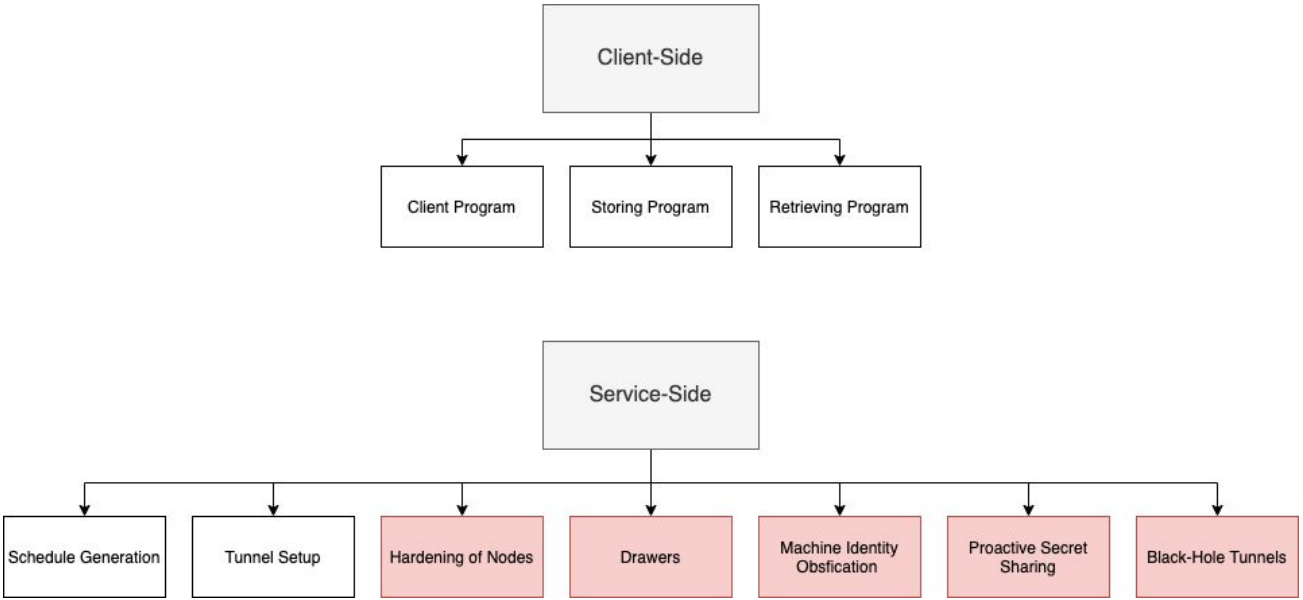
MAZE Overview



MAZE System

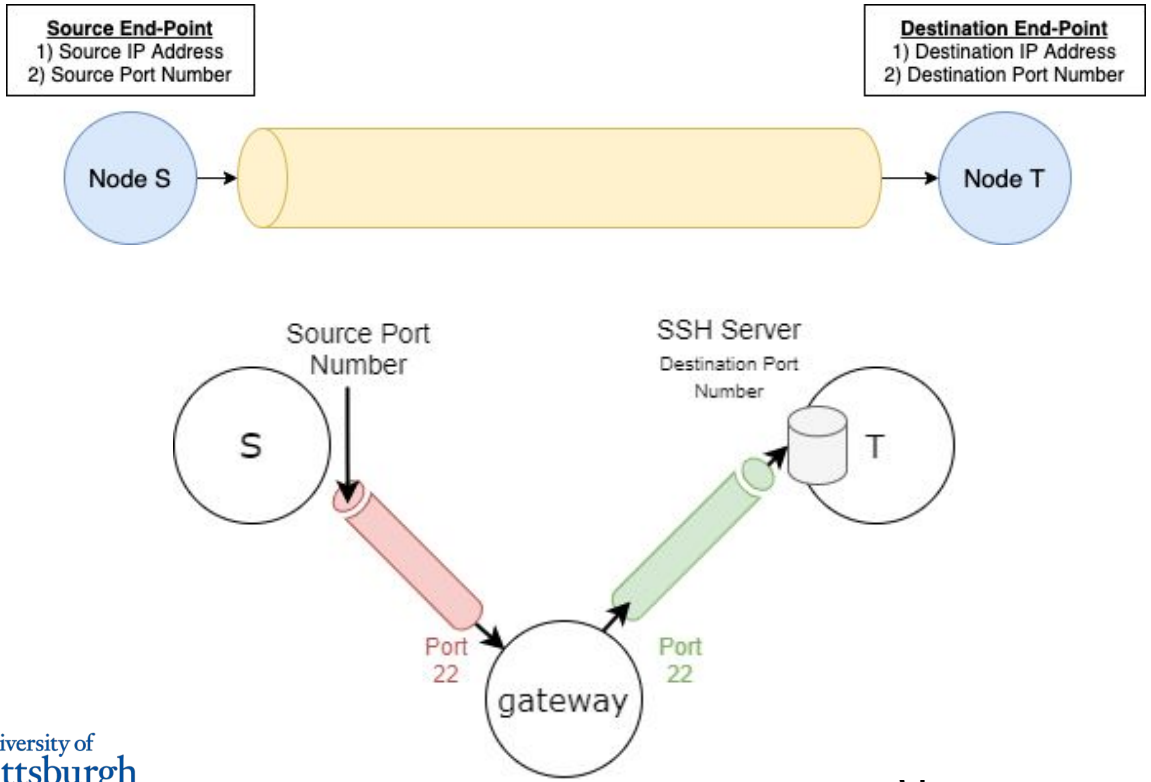

Gateway node
Client node
File node
File piece
Networking Link
Client Node

MAZE Design



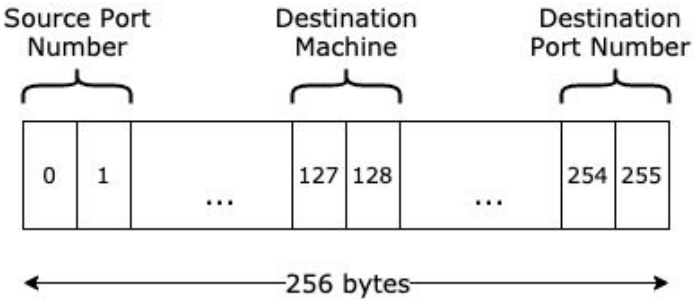
boxes in red are not currently implemented

MAZE Design: Tunnel



MAZE Design: Service-Side

- Generate the **schedule**
  - Creating an array of **consecutive** tunnels for a client to traverse in order to store or retrieve a file
  - End-points of the tunnel are determined by hashing a password



- Setup the **tunnels**
  - Iterate through the **schedule** and establish the respective tunnels
  - Start SSH servers at destination end-points

MAZE Design: Service-Side

- Hardening of nodes
  - Cost of traversing a regular networking link > Cost of traversing a tunnel
  - Limit connections to tunnel end-points; only accept connections from **localhost**
- Refresh Periods
  - (1) Nodes are restarted and system software is copied from a secure read-only medium
  - (2) File pieces are modified using proactive secret sharing
- Black-Hole Tunnels
  - Tunnels that lead to nodes that do not store any file and have no outgoing tunnels

MAZE Design: Client-Side

- **Command-line program** for interacting with MAZE
  - Accepts user parameters (path to file, password, and number of tunnels)
  - **"$ maze -s keys.txt -p pittsburgh -n 10"**
- Automated agent follows the schedule to **store** or **retrieve** a file
  - Connect to the respective **source port numbers** to arrive at the subsequent node

Experiments (1/2)

- Goal of the experiments
  - Evaluate the overhead of MAZE compared to directly using ***scp*** to transfer (i.e., store and retrieve) the files

$$\frac{\text{average time with MAZE - average time without MAZE}}{\text{average time without MAZE}}$$

- Measure overhead by varying **file sizes** and **number of tunnels**
  - We measured the average time (in secs) of 10 runs

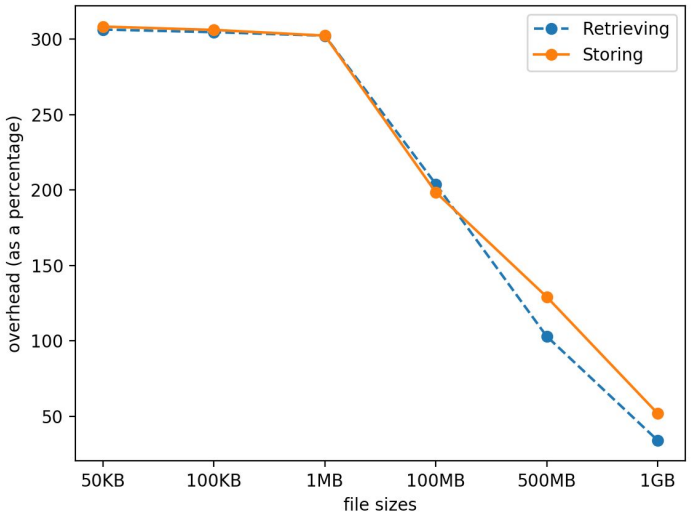
Experiments (2/2)

- Experiment Setup
  - 12 t2.micro Amazon EC2 instances with 8GB of memory running Ubuntu 16.04
  - Setup one instance as the **gateway node,** one instance as the **client node**, the remaining ten as **file nodes**

Instance	vCPU*	CPU Credits / hour	Mem (GiB)	Storage	Network Performance
t2.nano	1	3	0.5	EBS-Only	Low
t2.micro	1	6	1	EBS-Only	Low to Moderate
t2.small	1	12	2	EBS-Only	Low to Moderate
t2.medium	2	24	4	EBS-Only	Low to Moderate
t2.large	2	36	8	EBS-Only	Low to Moderate
t2.xlarge	4	54	16	EBS-Only	Moderate
t2.2xlarge	8	81	32	EBS-Only	Moderate

Results (varying file sizes)

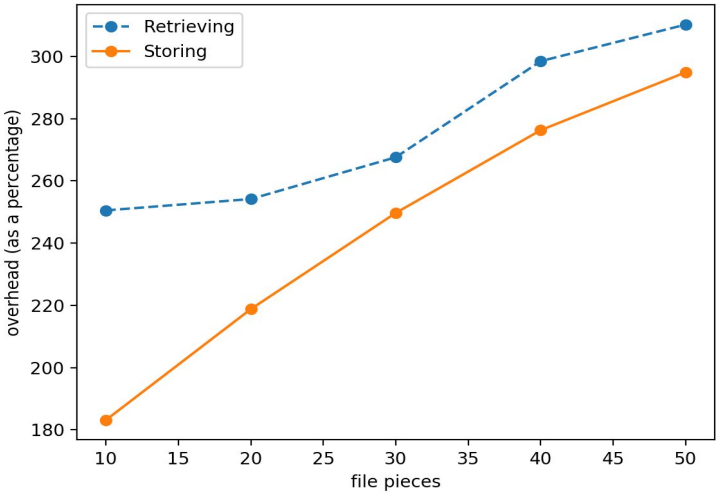
varying files sizes with a constant number of 10 tunnels



Average Storing Time (secs)		
file size	with MAZE	without MAZE
50KB	12.179	2.982
100KB	12.146	2.990
1MB	12.186	3.028
100MB	12.358	4.139
500MB	19.046	8.308
1GB	26.483	17.424
Average Retrieval Time (secs)		
file size	with MAZE	without MAZE
50KB	12.170	3.004
100KB	12.158	2.994
1MB	12.213	3.035
100MB	12.309	4.049
500MB	16.579	8.164
1GB	23.445	17.481

Results (varying file pieces)

varying file pieces with a constant file size of 100MB



Average Storing Time (secs)		
file pieces	with MAZE	without MAZE
10	12.420	4.387
20	24.852	7.796
30	39.212	11.213
40	54.933	14.598
50	72.13	18.265
Average Retrieval Time (secs)		
file pieces	with MAZE	without MAZE
10	12.345	3.171
20	24.891	7.028
30	38.925	10.589
40	56.163	14.094
50	71.941	17.536

Limitations

Hash Collisions
- Unreserved TCP port numbers range from 1,024 to 65,535
- Keep a list of tunnels that want to established at a certain source port number

- Performance Overhead
  - Primary bottleneck is the time needed to setup the tunnels
  - Tradeoff between performance and security

- Single Gateway Machine
  - Susceptible to Denial of Service Attacks

Future Work

- Complete implementation and evaluation of MAZE
  - Hardening of Nodes, Black-hole tunnels, Node refreshes, and Proactive Secret Sharing
- Build an attack simulator
  - Develop intelligent attackers that attempt to determine the end-points of the tunnels through traffic analysis

Questions?